CLAIMS

1. A method of making instructions of an electronic portable object XμP secure, which object is executing a program P supplied by a non-secure other electronic object XT in the form of a succession of F instructions, F thus denoting the number of instructions of said program P, said method using:

- a secret-key protocol co-operating with an ephemeral secret key K;

- a symmetrical cryptographic MAC function μ_K co-operating with a hash function $HASH_1$ defined by a compression function H_1 and a constant IV_1, and with a hash function $HASH_2$ defined by a compression function H_2 and a constant IV_2; and

- a program identifier ID stored in the electronic object XμP and corresponding to hashing of P;

said method being characterized in that said public-key protocol comprises the following stages:

a) an initialization stage during which the XμP generates an ephemeral key K, then receives from the XT the set of programs P, the number of instructions F and its identifier ID, computes the hash h of said program P with the $HASH_1$ function, by using the compression function H_1 and the constant IV_1, and finally generates signatures σ_i, by means of the μ_K function and of the key K, which signatures σ_i it transmits to the XT;

b) an execution phase during which the XμP checks that h and ID are equal, also verifies that ID is stored in its non-volatile memory, and then requests, one after the other, the instructions of P so as to execute them, and, for some of them, performs a sub-stage of verification that consists in requesting a signature σ, constructed on the basis of the signatures σ_i generated during the initialization stage and by means of the $HASH_2$ function, and in verifying said signature σ;

c) a reaction stage that takes place whenever a signature σ is not valid.

2. A method of making instructions of an electronic portable object secure according to claim 1, characterized in that the sub-stage of verification in the execution stage is verification of the signature σ taking place prior to execution of each instruction.

3. A method of making instructions of an electronic portable object secure according to claim 2, characterized in that the execution stage comprises the following sub-stages:

b-1) the XμP requests an instruction from the XT;

b-2) the XμP requests a signature σ constructed on the basis of the signatures σ_i generated during the initialization stage and by means of the $HASH_2$ function, and, in the event that said signature σ is not valid, executes the reaction stage; and

b-3) the XμP executes the instruction and returns to the sub-stage b-1.

4. A method of making instructions of an electronic portable object secure according to claim 1,

characterized in that the sub-stage of verification in the execution stage is verification of the signature σ taking place prior to execution of the instruction, if said instruction is an instruction that is critical for security.

5. A method of making instructions of an electronic portable object secure according to claim 4, characterized in that the execution stage comprises the following sub-stages:

b-1) the XμP requests an instruction from the XT;

b-2) if said instruction is critical for security, the XμP requests a signature σ constructed on the basis of the signatures σ_i generated during the initialization stage and by means of the $HASH_2$ function, and, in the event that said signature σ is not valid, executes the reaction stage; and

b-3) the XμP executes the instruction and returns to the sub-stage b-1.

6. A method of making an electronic portable object secure according to claim 1, characterized in that the sub-stage of verification in the execution stage is verification of the signature σ taking place prior to execution of the instruction if said instruction is an instruction that is critical for security, and if at least one of the items of data used for said instruction is a secret item of data.

7. A method of making instructions of an electronic portable object secure according to claim 6, characterized in that it uses a variable Φ defining the

set of security levels defined at a given instant by execution of a given program P, and in that the execution stage comprises the following sub-stages:

b-1) the XμP requests an instruction from the XT;

b-2) if said instruction is critical for security and if at least one of the items of data used by the instruction is secret, then the XμP requests a signature σ constructed on the basis of the signatures σ_i generated during the initialization stage and by means of the $HASH_2$ function, and, in the event that said signature σ is not valid, executes the reaction stage; and

b-3) the XμP executes the instruction, updates the security level (secret or non-secret data) of each of the items of data coming from the execution, and returns to the sub-stage b-1.

8. A method of making instructions of an electronic portable object secure according to claim 7, characterized in that it uses a variable Φ defining the set of security levels defined at a given instant by execution of a given program P, in that it uses an Alert Boolean function, and in that the execution stage comprises the following sub-stages:

b-1) the XμP requests an instruction from the XT;

b-2) if said instruction is critical for security and if the Alert Boolean function determined on the basis of the security level of the data used by the instruction and by the nature of the instruction itself is evaluated as TRUE, then the XμP requests a signature

σ constructed on the basis of the signatures σ_i generated during the initialization stage and by means of the $HASH_2$ function, and, in the event that said signature σ is not valid, executes the reaction stage; and

b-3) the XμP executes the instruction, updates the security level (secret or non-secret data) of each of the items of data coming from the execution, and returns to the sub-stage b-1.

9. A method of making instructions of an electronic portable object secure according to claim 1, characterized in that it uses a $HASH_3$ function defined by a compression function H_3 and a constant IV_3, and in that the program P is supplied in the form of a succession of G sections or blocks of instructions, G thus denoting the number of sections of said program.

10. A method of making instructions of an electronic portable object according to claim 9, characterized in that said protocol comprises the following stages:

a) an initialization stage during which the XμP generates an ephemeral key K, then receives from the XT the entire set of the program P, its number of sections G and its identifier ID, computes the hash h of said program P with the $HASH_1$ function, by using the compression function H_1 and the constant IV_1, and with the $HASH_3$ function, by using the compression function H_3 and the constant IV_3, and finally generates signatures σ_j, by means of the μ_K function and of the key K, which signatures σ_j it transmits to the XT;

b) an execution phase during which the XμP checks that h and ID are equal, also verifies that ID is stored in its non-volatile memory, and then requests, one after the other, the sections of P so as to execute them, and, for some of them, performs a sub-stage of verification that said sections comply, and then finally, for the final instruction of certain sections, performs a sub-stage of verification that consists in requesting a signature σ, constructed on the basis of the signatures σ_i generated during the initialization stage and by means of the $HASH_2$ function, and in verifying said signature; and

c) a reaction stage that takes place whenever a signature σ is not valid or whenever a section does not comply.

11. A method of making instructions of an electronic portable object secure according to claim 10, characterized in that the sub-stage of verification that a given section complies consists in verifying that no instruction of that section, except possibly for the last instruction, is an instruction that is critical for security.

12. A method of making instructions of an electronic portable object secure according to claim 11, characterized in that the sub-stage of verification in the execution stage is verification of the signature σ taking place prior to execution of the final instruction of each section.

13. A method of making instructions of an electronic portable object secure according to claim 12, characterized in that the execution stage comprises the following sub-stages:

b-1) the XμP requests a section from the XT;

b-2) for each non-final instruction of the requested section, the XμP verifies whether said instruction is critical, and, if it is, performs the reaction phase, and, otherwise, executes said instruction and goes to the next instruction;

b-3) for the final instruction of the requested section:

b-31) the XμP requests a signature σ constructed on the basis of the signatures σ_j generated during the initialization stage and by means of the $HASH_2$ function, and, in the event that said signature σ is not valid, executes the reaction stage; and

b-32) the XμP executes the instruction;

b-4) the XμP then returns to the sub-stage b-1.

14. A method of making instructions of an electronic portable object secure according to claim 11, characterized in that the sub-stage of verification in the execution stage is verification of the signature σ taking place prior to execution of the final instruction of each section, if said instruction is an instruction that is critical for security.

15. A method of making instructions of an electronic portable object secure according to claim

14, characterized in that the execution stage comprises the following sub-stages:

b-1) the XμP requests an instruction from the XT;

b-2) for each non-final instruction of the requested section, the XμP verifies whether said instruction is critical, in which case it performs the reaction stage, and otherwise it executes said instruction and goes on to the next instruction;

b-3) for the final instruction of the requested section:

b-31) if the instruction is critical for security, the XμP requests a signature σ constructed on the basis of the signatures σ_j generated during the initialization stage and by means of the $HASH_2$ function, and, in the event that said signature σ is not valid, executes the reaction stage; and

b-32) the XμP executes the instruction; and

b-4) the XμP then returns to the sub-stage b-1.

16. A method of making instructions of an electronic portable object secure according to claim 11, characterized in that the sub-stage of verification in the execution stage is verification of the signature σ taking place prior to execution of the final instruction of each section, if said instruction is an instruction that is critical for security, and if at least one of the items of data used by said instruction is a secret item of data.

17. A method of making instructions of an electronic portable object secure according to claim

16, characterized in that it uses a variable Φ defining the set of security levels defined at a given instant by execution of a given program, and in that the execution stage comprises the following sub-stages:

b-1) the XμP requests an instruction from the XT;

b-2) for each non-final instruction of the requested section, the XμP verifies whether said instruction is critical, in which case it performs the reaction stage, and otherwise it executes said instruction and goes on to the next instruction;

b-3) for the final instruction of the requested section:

b-31) if the instruction is critical for security, and if at least one of the items of data used by the instruction is secret, the XμP requests a signature σ constructed on the basis of the signatures σ_j generated during the initialization stage and by means of the $HASH_2$ function, and, in the event that said signature σ is not valid, executes the reaction stage; and

b-32) the XμP executes the instruction;

b-33) the XμP updates the security level (secret data or non-secret data) of each of the items of data coming from the execution; and

b-4) the XμP then returns to the sub-stage b-1.

18. A method of making instructions of an electronic portable object secure according to claim 16, characterized in that it uses a variable Φ defining the set of security levels defined at a given instant

by execution of a given program, in that it uses an Alert Boolean function and in that the execution stage comprises the following sub-stages:

b-1) the XμP requests an instruction from the XT;

b-2) for each non-final instruction of the requested section, the XμP verifies whether said instruction is critical, in which case it performs the reaction stage, and otherwise it executes said instruction and goes on to the next instruction;

b-3) for the final instruction of the requested section:

b-31) if the instruction is critical for security, and if the Alert Boolean function determined on the basis of the security level of the data used by the instruction and by the nature of the instruction itself is evaluated as being TRUE, the XμP requests a signature σ constructed on the basis of the signatures σ_j generated during the initialization stage and by means of the $HASH_2$ function, and, in the event that said signature σ is not valid, executes the reaction stage; and

b-32) the XμP executes the instruction;

b-33) the XμP updates the security level (secret data or non-secret data) of each of the data coming from the execution; and

b-4) the XμP then returns to the sub-stage b-1.

19. A method of making instructions of an electronic portable object secure according to any one of claims 4 to 8, or 11 to 18, characterized in that at

least one of the following types of instruction are critical for security:

- the test instructions and/or
- the instructions issuing information to the outside via communications means and/or
- the instructions modifying the contents of the non-volatile memory and/or
- the computation instructions presenting special cases during execution of them, such as the launch of exceptions.

20. A method of making instructions of an electronic portable object secure according to claim 8, or claim 18, characterized in that the Alert Boolean function is evaluated as being TRUE for at least one of the following types of instruction:

- the test instructions and/or
- the instructions issuing information to the outside via communications means and/or
- the instructions modifying the contents of the non-volatile memory and/or
- the computation instructions presenting special cases during execution of them, such as the launch of exceptions.

21. A method of making instructions of an electronic portable object secure according to claim 8, or claim 18, characterized in that the Alert Boolean function is evaluated as being TRUE for at least one of the following types of instruction, if at least one of the input items of data is secret, and as being FALSE if all of the items of data tested are public:

- the test instructions and/or

- the instructions issuing information to the outside via communications means and/or

- the instructions modifying the contents of the non-volatile memory and/or

- the computation instructions presenting special cases during execution of them, such as the launch of exceptions.

22. A method of making instructions of an electronic portable object secure according to any one of claims 7 or 8, or 17 or 18, characterized in that the set of security levels Φ used during execution of a program P is indicated by the value of a function φ, such that, for any item of data u used by the program, $\varphi(u)=0$ designates the fact that u is public and $\varphi(u)=1$ designates the fact that u is private, and such that, for any item of data v resulting from execution of an instruction of the program P, $\varphi(v)=1$ if at least one of the items of input data of the instruction is private, and, otherwise $\varphi(v)=0$.

23. A method of making instructions of an electronic portable object secure according to claim 22, characterized in that the values of the function φ are computed by means of hardware implementation of a "Logic OR" function implemented on the values of the φ function for the input data of the instructions.

24. A method of making instructions of an electronic portable object secure according to any one

of claims 1 to 23, characterized in that the hash functions $HASH_1$, $HASH_2$, and $HASH_3$ are identical.

25. An electronic object, characterized in that it implements any one of claims 1 to 24.